Methods of Experimental Particle Physics

Alexei Safonov

Lecture #20
Deficiencies of Universal Generators

• “General purpose” generators are based on Leading Order (LO) QCD plus Parton Showering
  • As an approximation (e.g. breaking things up in stages and prohibiting interference effects is not fair), it tends to underestimate hardness of the events with multiple jets
    • E.g. W+1,2,3-Jets are important backgrounds to many signals, need to get them right in your background simulation
  • One improvement would be to explicitly allow diagrams with W+N partons instead of doing W+1 parton and hope that Parton Showering will do the rest for you
    • But then you still need to attach parton showering to turn those final partons you just created into jets
Double Counting

• The problem arises as e.g. W+1parton+PS will actually make multiple parton final states by making extra partons through showering (it just doesn’t make enough of them as it misses some terms)
  • If you choose to do N partons “by hand” instead, you have done at least part of the job of what the PS is doing in the “general purpose” case
  • So what you want is “undo” the making hard partons by PS (as you think it’s not quite correctly done), replace it with your calculation “by hand” and keep everything else that the PS did afterwards
  • It’s clear that doing things right may not be trivial
    • Definitely you can’t just run PS on top of W+4partons pQCD calculation as it will re-do some of the extra parton generation again, which you don’t trust it with
• Special “matching algorithms” needed
N Parton ME Generators

- MadGraph, AlpGen are examples of ME (= Matrix Element) generators
  - Allows you just directly calculate tree-level diagrams with say \( W+4 \) partons using normal pQCD methods
    - This is not really NLO as NLO would also include all sorts of loop diagrams etc.
    - Can’t use it “as is” either, because it’s missing a lot of things, e.g. it completely ignores collinear divergences by pretending they don’t exist (LLA-based Parton Showering is getting that done correctly)
    - Also can’t compare to data, someone still needs to do fragmentation of those partons
  - So at the end it’s sort of like a special trick designed to correct deficiencies of PS by doing this chain:
    - Use “ME generator” to calculate \( W+4 \) partons and make events
    - Run Parton Showering (from Pythia) on these events
    - Remove the “overlap” part of phase space
Matching Schemes

Event Generators for Many Hard Partons

- Want to use these matrix-element tools with parton showers
- Each topology (e.g. W + 0, 1, 2, 3, 4 partons) has no soft/collinear approximation
- How do I rigorously add a parton shower to each topology with no double counting of hard emissions?

Solution (CKKW):

1. Make the $|\mathcal{M}|^2$ result “look” like a parton shower down to a reasonable cutoff scale ($k_T^{\text{cut}}/Q_{\text{hard}} \sim 0.1$)
2. Add on ordinary parton shower below $k_T^{\text{cut}}$
Matching Algorithm Example

Review of Matching

Pseudo-Shower Method

1. Generate \( W + N \) parton events, applying a cut \( p_{T\text{cut}}^2 \) on shower \( p_{T}^2 \) (\( p_{T}^2 \) for ISR, \( z(1-z)m^2 \) for FSR)

2. Form a \( p_{T}^2 \)-ordered parton shower history

3. Reweight with \( \alpha_s(p_{T}^2) \) for each emission

4. Add parton shower and keep if no emission harder than \( p_{T\text{cut}}^2 \): (save the first event with full topology)

5. Remove softest of \( N \) partons, fix up kinematics, add parton shower and keep if no emission harder than \( p_{T\text{softest}}^2 \)

6. Go to 4 until no partons remain, or an emission is too hard

7. If not rejected, use the saved event
Can One Do Better?

• ME plus Matching works well
  • But as we agreed it is a patch or a trick and not a general solution

• MC@NLO is an example of true NLO order generator, which is doing things properly
  • Not easy, as consistent use of pQCD (hard emissions) and PS (soft emissions) is not trivial
    • E.g. various divergences actually cancel out, but what would you do if infinities appear in different parts of the code?
      • Need to jump through various hoops to re-arrange things in order to ensure cancellation of divergences
      • E.g. see Frixione and Webber “Modified Subtraction Method”

• So it’s possible, it’s just very difficult due to poor QCD collinear/infrared behavior
Getting Detector Into the Game

- Everything discussed so far pretended that there is no detector, everything is perfect
- Given that we already have “events” why not make one more step and make things even more convenient by including detector effects, e.g.
  - “Smear” true particle energies with detector resolutions to emulate detector’s imperfect response
  - Account for overlaps of energy deposits by nearby particles
  - Take into account energy losses (an electron hitting the calorimeter has likely already lost some of it energy as it was getting through the tracker, especially if the tracker is “heavy” like it is at CMS
  - Take into account effects of imperfect algorithms in event selection (trigger) which likely lead to reduced resolution and accuracy
Propagation Through Matter

- Even better if we could package all this emulated information into the same data format as in real data
- Very convenient as you will be able to run the same analysis code on data and on simulation without any (or almost any changes) to it
  - One can also make direct comparisons, find anything imperfect and fix
- A key element required is code capable of propagating particles through the detector material
  - Note it’s not just a matter of convenience as you need it to design detectors in the first place
GEANT

• GEANT is a software package designed to propagate individual particles through matter and emulate energy loss
  • E.g. in designing a gas detector system, you would define “active volume” and specify the gas, pressure etc.
    • GEANT will calculate $dE/dx$ and simulate energy loss by a charged particle of this type/energy
      • Probabilistically, i.e. by tossing a coin as we do often in Monte Carlo calculations
  • By defining all other elements of your detector, including support structures etc and making GEANT propagate through all of the material, you will simulate energy loss prior to arrival to your detector
GEANT

• A package used across various sub-fields of particle physics as well as in nuclear, material sciences, medical applications etc.
  • Enormous amount of validation and verifications to ensure ever improving parameterizations in describing particle interactions with material
  • Truly global effort with many contributors
Detector Response

- GEANT only tells you that there was an energy release (an ionization ion/electron with known energy)
- Someone still needs to emulate how that ionization leads to an electrical signal:
  - Other programs (e.g. Garfield) can be used to emulate avalanche development in electric (and magnetic if it’s there) field to predict when (time) and how much electric charge would arrive to a particular point
    - Presumably you would want to know that for the point where your anode or cathode is if you want to measure charge or current
  - You also need to emulate response of your electronics (signal amplitude, shape versus time, any digitization or threshold/discriminator effects)
    - Note this is likely exactly the same information your actual detector is using and this information is likely encoded in some data structures that are later used in reconstruction. If you encode this simulated information in the data format, you can use your standard reconstruction methods on the simulated data
Simulation at Actual Experiments

- CDF actively used various parameterized response functions (based on test beam studies and GEANT)
  - A way to save CPU, which was a substantial problem in the past
- CMS simulation uses GEANT pretty much everywhere
  - It also has a faster parameterized simulation, but a full blown GEANT version is still widely used and is the default
- The degree of detail to which detector response is emulated varies system by system within a single experiment
  - CMS in general though has much more detailed and faithful emulations, including electronics and trigger algorithms, then say CDF
  - Of the two endcap muon systems, CSC and RPC, the CSC simulation is substantially more detailed and elaborate
- Generally, the level of agreement between the background data and simulation is a function of time and effort put into simulation
  - CMS simulation is much better reproducing the data compared to CDF, but largely because of an order of magnitude more effort put in that:
    - Careful and very detailed material description including both active and passive material, all sorts of shielding etc.
    - Incredibly detailed replication of many electronics and software algorithms in simulation
CDF Examples: Calorimeter Measurements

CDF Run 2 Preliminary

\[ \beta = \frac{P_T^{\text{probe}}}{P_T^{\text{trig}}} \]

**Jet \eta**

- **Jet50 data 5.3.1pre2**
- **dijet50 MC 5.3.1pre2**

Rest of the towers: EoP (HA) >0.5 GeV

**CDF Run II Preliminary**

\[ \Psi(r) = \frac{1}{N_{\text{jets}}} \sum_{\text{jets}} \frac{P_T(0,r)}{P_T(0,R)} \]

- **Z \rightarrow e^+e^- + jets**
- **Data**
- **Pythia Tune A**
- **Aipgen+Herwig**
- **MadGraph+Pythia Default**

Uncorrected
- Statistical errors only
**Example: CMS GEM Simulation**

- Gaseous detector capable of operating at very high rates
  - An R&D project proposed to improve CMS muon system
  - A lot of simulation studies ongoing to aid in the design

Geometry of the active volumes defined and passed to GEANT in a special format

Typical ionization loss for an energetic muon flying through GEM as predicted by GEANT
Beyond GEANT

• GEANT4 describes interaction of particles with the material
• It does not know anything about how your detector works
  • Electric field, avalanche shape, timing, or magnitude of the electrical signals collected on the readout elements
    • GARFIELD program can do that (see the illustration showing avalanche formation)

• These effects can be studied using dedicated tools to develop a parameterized model to be implemented for converting GEANT “SimHits” into simulated signals (electrical currents) on sensitive elements of the detector (strips in the case of GEMs)
• Use results to create a parameterized model that in the simulation will be taking GEANT hits, emulating electrical response of the readout, including various electronics digitization effects
Next Time

• Statistics
  • Parameter measurements and limits
    • Frequentist and Bayesian
  • Hypothesis testing